def Moves(CL):

moves = []

if (CL[2] == 'WHITE' and CL[3] == 'PAWN'):

#пусть наша пешка стоит на поле A2, нам надо проверить поле A3

y = CL[1]+1

coord = [CL[0], y]

changes = []

CL\_NEW = Cell.A1(coord, changes)

if (CL\_NEW[3] == ''):

moves.append([CL[0], CL[1], CL\_NEW[0], CL\_NEW[1]])

Пусть наша пешка стоит на поле А2. По правилам, если поле А3 пустое, то пешка может пойти на него. И если поле А4 также пустое, пешка также может на него пойти, при условии, что она до этого стояла на второй горизонтали.

Мы уже проверили, что А3 – пустое. Теперь надо проверить, что А4 тоже пустое и пешка стояла на А2.

По сути, нам надо добавить новое условие в последнее условие текущего состояния функции.

Нам надо вызвать функцию по новым координатам.

ОТЛИЧНО!

Мы научили комп двигать пешку на клетку вперёд, если впереди пусто. И на две клетки вперёд, если впереди пусто.

Проверить:

1. Ходит с А3
2. Если А3 занято

Проверено. Работает.

Нам надо научить компьютер понимать, что он может побить вражескую фигуру наискосок.

Значит, нам нужно условие, которое будет проверять, во-первых, что такое поле СУЩЕСТВУЕТ. Во-вторых, что клетки наискосок можно бить. Это делается путём проверки цвета, то есть CL\_NEW[2] == ‘ENEMY\_COLOR’.